



Attorney Docket No. GB920000078US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Anthony J. O'Dowd
Docket No.: GB920000078US1
Serial No.: 09/682,520
Filing Date: September 13, 2001
Group: 2193
Examiner: Trenton J. Roche

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature: *Linda L. Vulpis*

Date: June 28, 2006

Title: Tracing the Execution Path of a Computer Program

REPLY BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

Sir:

Appellants present the following remarks addressing the arguments raised in the Examiner's Answer dated April 28, 2006. Appellants incorporate herein all remarks previously presented, including those presented in Appellants' Appeal Brief dated October 5, 2005.

ARGUMENTS

The Examiner's Answer reiterates many of the arguments previously raised by the Examiner and which are fully addressed in Appellant's Appeal Brief. Thus, the present Reply Brief addresses the Examiner's latest comments in the "Response to Arguments" section of the Examiner's Answer (pages 9-13).

To reiterate for the convenience of the reader, independent claim 1 recites a method for tracing the execution path of a computer program comprising at least one module including a plurality of instructions, at least one of said instructions being a branch instruction, the method comprising the steps of identifying each branch instruction, evaluating each branch instruction to be one of true and false, and responsive to an evaluation of true, pushing a unique identifier into a

predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true. Claim 13 recites, *inter alia*, a pusher, responsive to an evaluation of true, for pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true. Claim 14 recites, *inter alia*, the step of instrumenting said instructions associated with an evaluation of true with a signature instruction, wherein said signature instruction causes a unique identifier to be pushed into a predefined area of storage upon execution of said true instructions at run-time.

In accordance with an illustrative embodiment explained in the present specification, at paragraphs 48 and 49, upon execution of the program, a small, fixed size area called a signature area is defined. The signature area may contain up to a certain number of signature points. Each signature point comprises a unique 4 bit identifier. This identifier is, according to this illustrative embodiment, used to indicate the execution path or flow followed through the program. According to this illustrative embodiment, signature points are added to the signature area. For example, as explained at paragraph 57 of the present specification with respect to FIG. 5, case statement 1 is evaluated to TRUE such that 1 is pushed into the signature area 300. Execution then jumps to case statement 4, case statement 2, case statement 3, case statement 1, and finally to case statement 2. Accordingly, the corresponding identifiers are pushed into the signature area (1, 4, 2, 3, 1, 2). These numbers indicate which set of instructions have been executed at run-time and in what order. The signature information provides valuable insight into the behavior of the program. Should the program fail or behave erroneously, then the signature points can be used in subsequent problem diagnostics (paragraph 58).

The Examiner continues to assert that Wisor discloses all the features of the claimed invention. Appellant continues to respectfully assert that this is incorrect. Wisor does not disclose “pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true,” as recited in claim 1; “a pusher, responsive to an evaluation of true, for pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true,” as recited in claim 13; or “instrumenting said instructions associated with

an evaluation of true with a signature instruction, wherein said signature instruction causes a unique identifier to be pushed into a predefined area of storage upon execution of said true instructions at run-time,” as recited in claim 14.

The Examiner, at page 10 of the Examiner’s Answer, states that he “has interpreted the ability of Wisor to populate a branch trace history buffer (BTHB) with individual bits representing the taken or not-taken status of branches to equate to the required limitation of ‘pushing a unique identifier into a predefined area of storage [wherein said unique identifier is associated with the instructions executed] as a result of said evaluation of true.’” The individual bits the Examiner refers to in Wisor are the logic ‘1’ value or the logic ‘0’ value that is inserted in the bitmap corresponding to a “taken” status and a “not-taken” status, respectively.

The Examiner goes on to state (bottom of page 11 and top of page 12) that there are two reasonable interpretations as to why these constitute unique identifiers:

1. The individual 1 and 0 bits inserted into the bitmap entries are unique in that each 1 or 0 inserted into a specific bitmap entry in the BTHB is assigned a specific bit position in the bitmap entry. Referring again to Figure 3, by this interpretation, the “1” inserted into bit position 26 is unique from other “1’s” in that it is the only “1” inserted into bit position 26 of that specific bitmap entry in the BTHB.
2. The individual 1 and 0 bits inserted into the bitmap entries are unique in that each 1 or 0 inserted into a specific bitmap entry in the BTHB (i.e. in the order that the branches are encountered in the program) is uniquely associated with the specific conditional branch it represents. Referring again to Figure 3, by this interpretation, the “1” inserted into bit position 26 is unique from other “1’s” in that it is the only “1” which corresponds to the specific branch that was taken at the time of the evaluation, and bit position 27 would represent a completely different branch from bit position 26.

With all due respect to the Examiner, these interpretations clearly illustrate the inaccuracy of the argument. The first interpretation offered by the Examiner does not illustrate that the identifier (which the Examiner clearly acknowledges are the ‘1’ bit or the ‘0’ bit) is unique but rather illustrates that the bit positions may be unique. Likewise, the second interpretation offered by the Examiner does not illustrate that the identifier is unique but rather illustrates that the corresponding branch may be unique. Again, it is clear that two bitmap entries referring to “taken”

conditional branches in Wisor may be the same and, therefore, not unique since both conditional branches would have the same bit set to '1' to indicate it is a conditional branch that is "taken" (as opposed to "not taken").

The fact that Wisor does not teach or suggest a "unique identifier" as in the claimed invention is further supported by the description at column 3, line 57, through column 4, line 9, of Wisor:

If a conditional branch is encountered, the debug software reads the next bit in the corresponding bitmap entry. If the previous trace event was not a conditional branch, the next entry in the BTHB should be a bitmap entry. Each bitmap entry, however, may contain information corresponding to a number of conditional branches. The encountered branch corresponds to the first bit in the bitmap. If the first bit is a 1, the branch was taken. The debug software will determine the next instruction from the conditional branch instruction itself. If the first bit is a 0, the branch was not taken and the debug software can simply continue with the next instruction in program order. If the trace event which preceded the conditional branch was another conditional branch, then the debug software had already read a bitmap to determine whether that branch had been taken. If less than all the bits in the bitmap have been used to determine whether conditional branches have been taken, the next bit is used to determine whether the current branch was taken. If all of the bits have been used, the next entry (also a bitmap entry) will be used for the current branch.

Clearly, the reason why the "debug software" in Wisor must look at a previous bitmap entry and/or a next bitmap entry is because the entries are not unique. This problem is overcome by the claimed invention.

Therefore, to reiterate, nothing in Wisor teaches or suggests that the bitmap entry or creation of a bitmap entry may involve: "pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true," as recited in claim 1; "a pusher, responsive to an evaluation of true, for pushing a unique identifier into a predefined area of storage, wherein said unique identifier is associated with the instructions executed as a result of said evaluation of true," as recited in claim 13; or "instrumenting said instructions associated with an evaluation of true with a signature instruction,

wherein said signature instruction causes a unique identifier to be pushed into a predefined area of storage upon execution of said true instructions at run-time,” as recited in claim 14.

Lastly, even if one were to acknowledge *arguendo* that the ‘1’ bit or the ‘0’ bit becomes unique once it is in a specific bit position of the bitmap entry associated with a specific branch, the ‘1’ bit or the ‘0’ bit only becomes “unique” after it is inserted in the bit position. On the other hand, the claim recites “pushing a unique identifier.” Thus, the claimed identifier is unique before and during pushing. This would not be the case for the bits in Wisor since, while being inserted or pushed, the bits are not unique.

For at least these reasons, Appellant asserts that independent claims 1, 13 and 14 are patentable over Wisor.

Regarding the rejections of claims 2 and 3 under 35 U.S.C. §103(a) as being unpatentable over Wisor, and claims 4-12 and 15 under 35 U.S.C. §103(a) as being unpatentable over Wisor in view of U.S. Patent No. 6,353,924 to Ayers et al (hereinafter “Ayers”), Appellants believe the Appeal Brief sets out proper reasons for traversing such rejections.

Appellants have attached hereto (as “Attachment 1”), an Evidence Appendix and a Related Proceedings Appendix which can be attached to the end of the Appeal Brief as pages 15 and 16.

In view of the above, Appellant believes that claims 1-15 are in condition for allowance, and respectfully requests favorable reconsideration and withdrawal of the various rejections.

Respectfully submitted,



William E. Lewis
Attorney for Appellant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946

Date: June 28, 2006



Attorney Docket No. GB920000078US1

EVIDENCE APPENDIX

None.



Attorney Docket No. GB920000078US1

RELATED PROCEEDINGS APPENDIX

None.



Attorney Docket No. GB920000078US1 *AT* *IFW*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Anthony J. O'Dowd
Docket No.: GB920000078US1
Serial No.: 09/682,520
Filing Date: September 13, 2001
Group: 2193
Examiner: Trenton J. Roche

I hereby certify that this paper is being deposited on this date with the U.S. Postal Service as first class mail addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature: *Lisa L. Tulpis* Date: June 28, 2006

Title: Tracing the Execution Path of a Computer Program

TRANSMITTAL OF REPLY BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is the following document relating to the above-identified patent application:

(1) Reply Brief.

It is believed that there is no additional fee due in conjunction with the response. In the event of any non-payment or improper payment of a required fee, the Commissioner is hereby authorized to charge or to credit **International Business Machines Corporation Deposit Account No. 50-0510** as required to correct the error.

Respectfully submitted,

William E. Lewis

Date: June 28, 2006

William E. Lewis
Attorney for Applicant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946